Return To Agenda Page

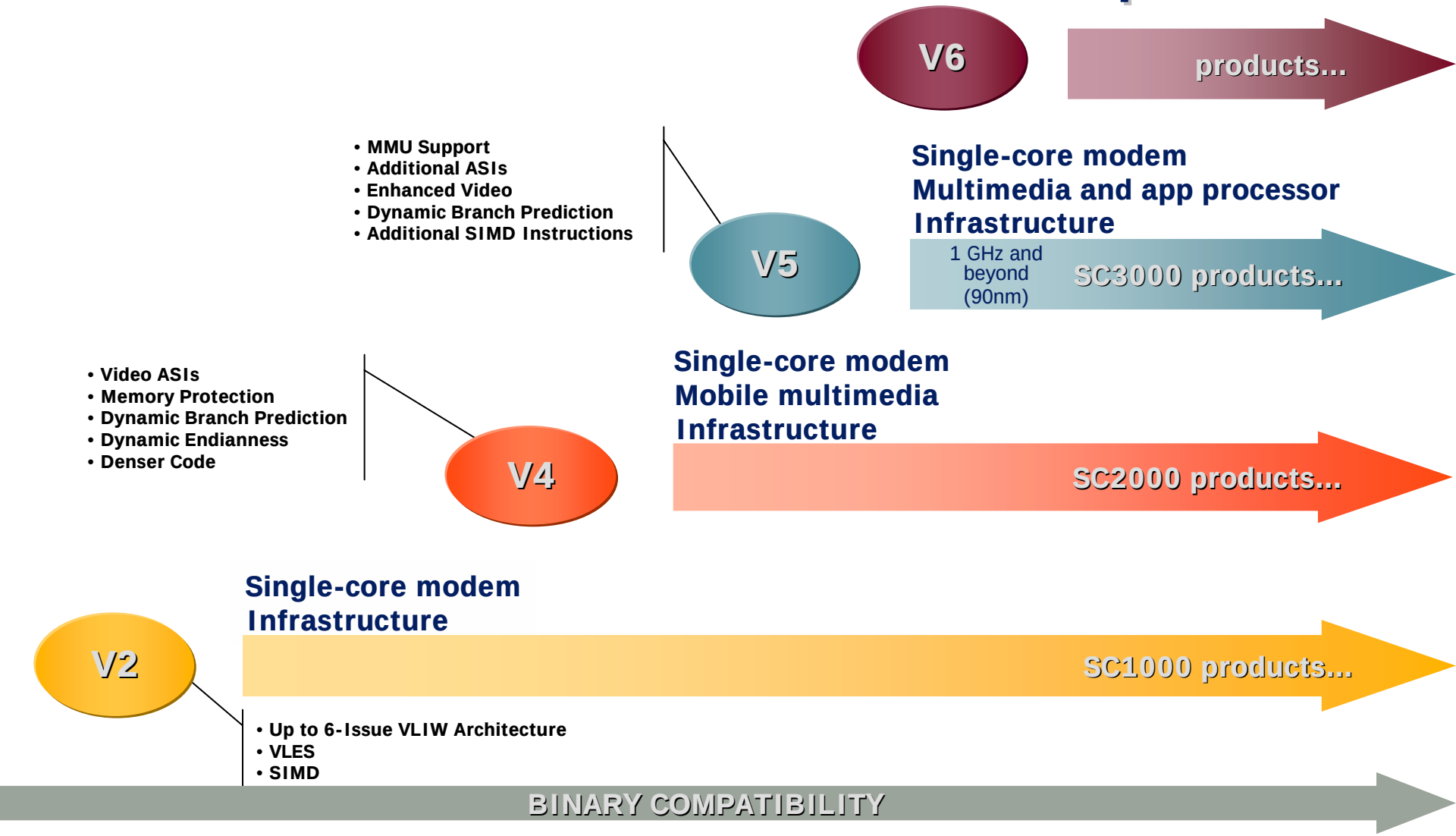
StarCore V5 Architecture

Amnon Rom, CTO & VP Engineering
StarCore LLC

Fall Processor Forum 2005

October, 2005

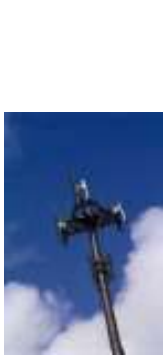
StarCore Architecture Roadmap



V1 and V3 have been developed and are being used by Freescale only

StarCore: A Versatile Architecture For Today's Market Challenges

- Wireless handsets
  - 2.5G, 3G (Motorola)
- Wireless base-station
  - Starlite® family from Freescale
- Cellular modems and multi-media applications
  - Skyworks, Motorola, Freescale
- VOIP phone application
  - Zultys Technologies, Legerity
- Mobile Extreme Convergence® (MXC)
  - Freescale, Motorola



MXC
"System in a Postage Stamp"

StarCore™

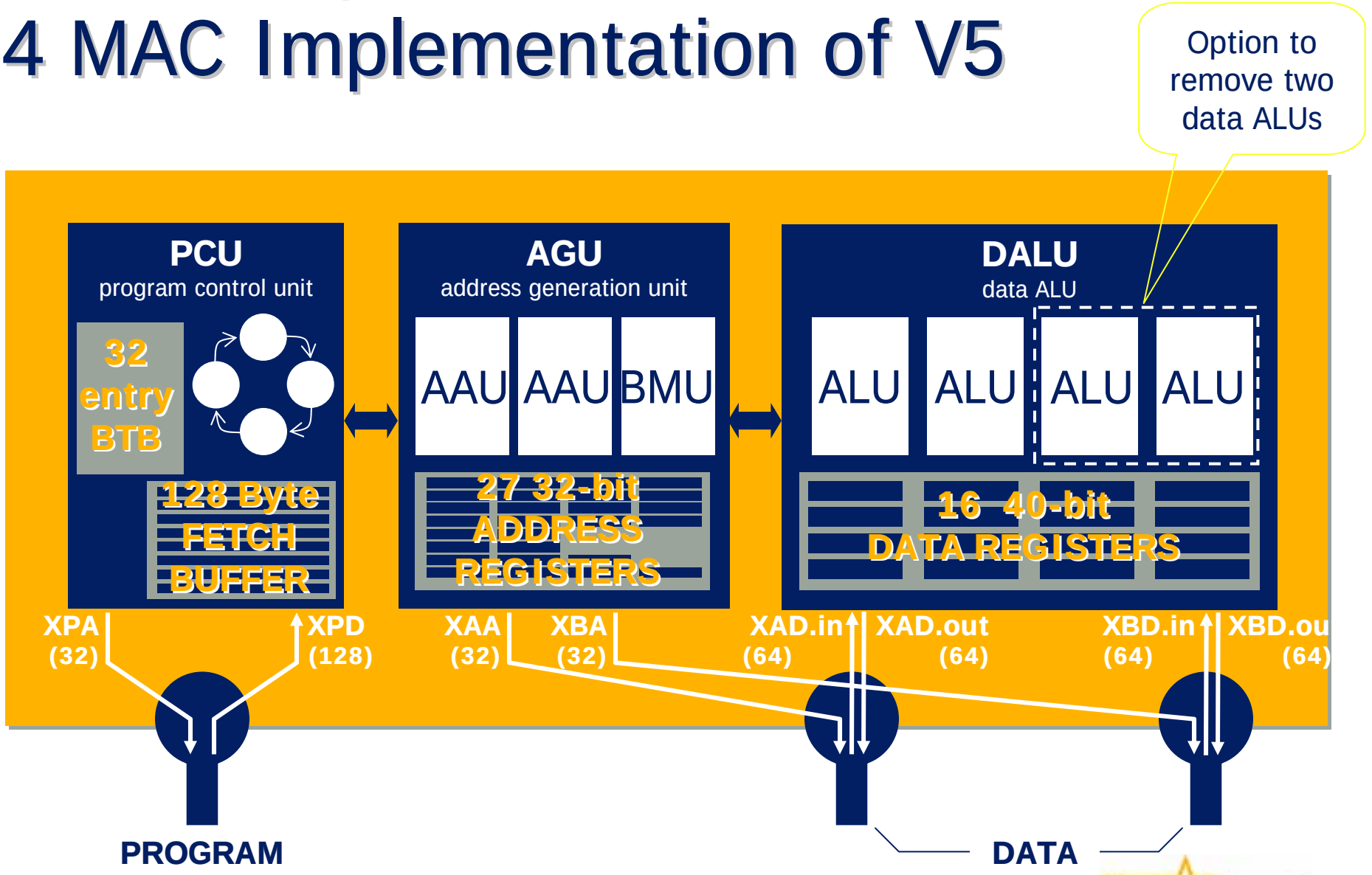
V5 Highlights

47 new instructions (relative to V4)	Better DSP performance Improved code density - comparable to ARM Thumb®
Enhanced multimedia and communication instructions	Efficient video codecs and modems in software
SIMD: 2 or 4 operands in a single ALU and application specific instructions	More effective multimedia and modem processing
Support for precise exceptions	Enable MMU for advanced OS support Improved error detection/correction for all memories
Wide stack operations	Fast context switch Better compiler support
Binary compatible with V2 (SC1200, and SC1400), V3 and V4 (SC2200, SC2400)	Protection of the customers' software investment

SC3000 Highlights - First Implementation of V5 Architecture

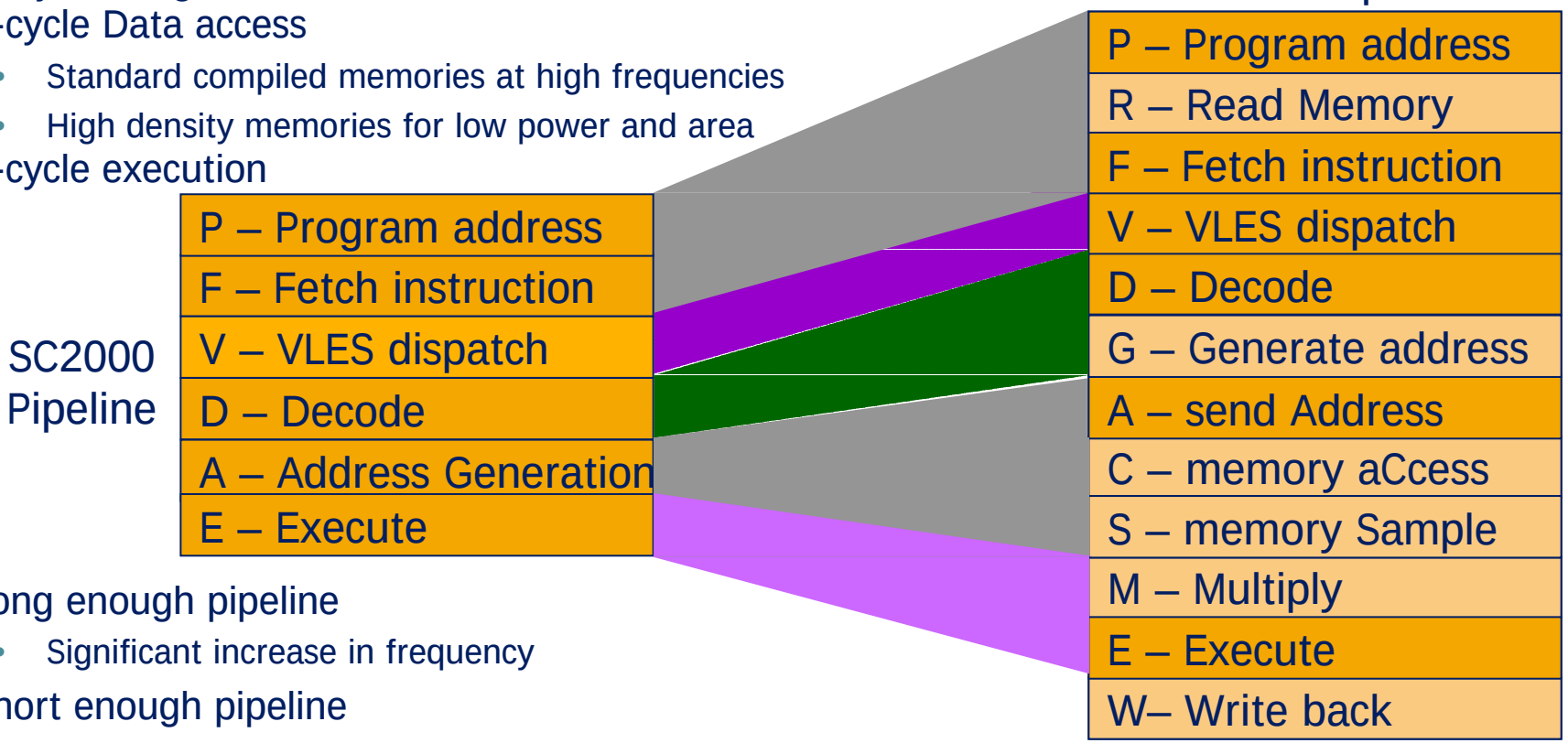
Significantly higher operating frequency than SC2000	Greater flexibility in applications processing
Very low power	Longer battery life Lower power per channel
Relaxed memory protocol	Interfaces to less expensive and denser memory for improved system integration
Fully interlocked	Much easier programming (no restrictions) Better C-compiler optimizations
Zero-latency load to execute Zero-latency execute to store	Good performance of legacy code Easier programming Reduces register pressure Improve compiler performance

Block Diagram – 4 MAC Implementation of V5



Twelve-Stage Pipeline of SC3000

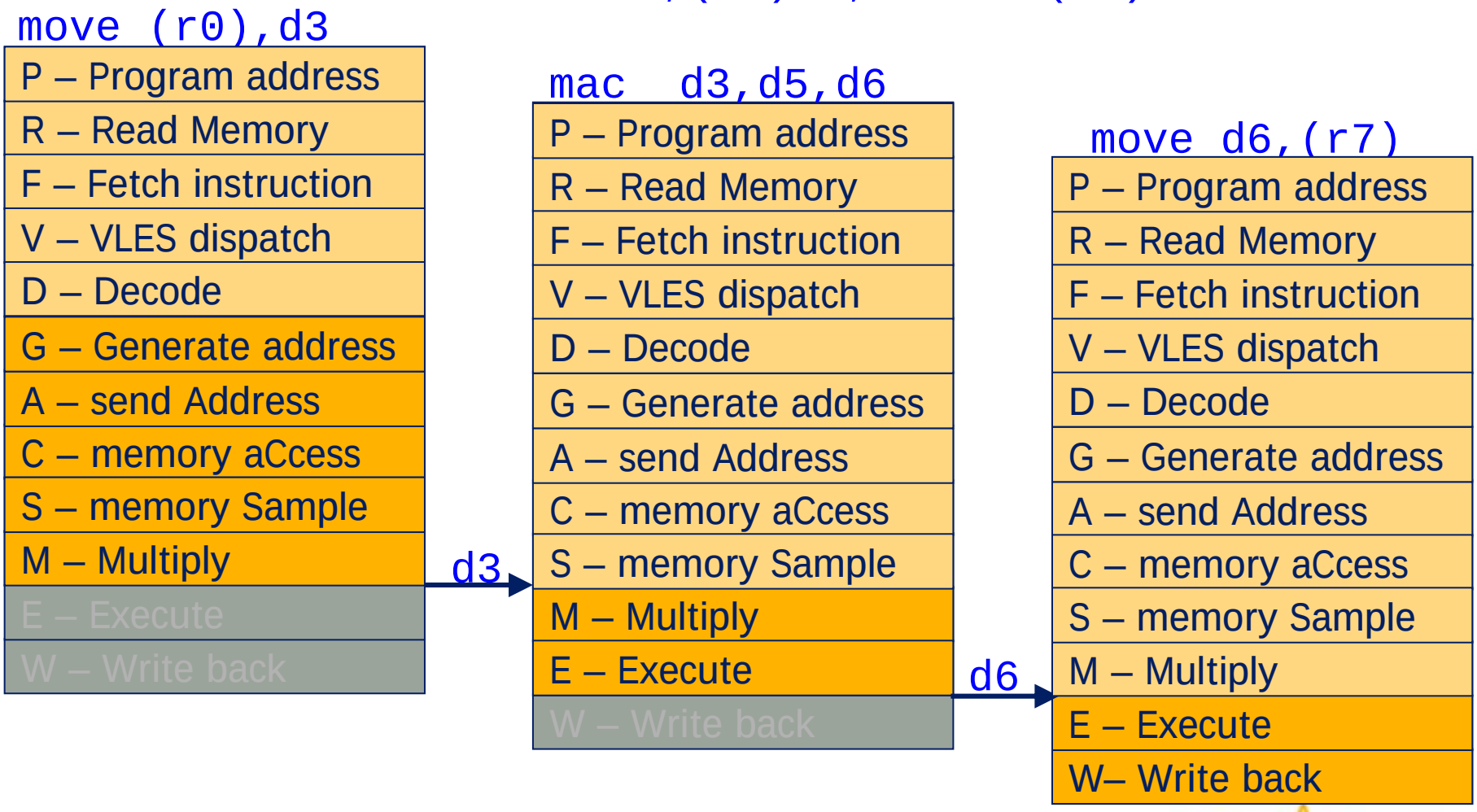
- 3-cycle Program access
- 4-cycle Data access
  - Standard compiled memories at high frequencies
  - High density memories for low power and area
- 2-cycle execution



- Long enough pipeline
  - Significant increase in frequency
- Short enough pipeline
  - Excellent legacy code performance
  - Good change of flow and interrupt performance
  - Zero latency load-execute-store
- Fully interlocked
  - Easier programming
  - Very good compiler target (improved performance and reduced code size)

Zero Latency Load-Execute-Store Example

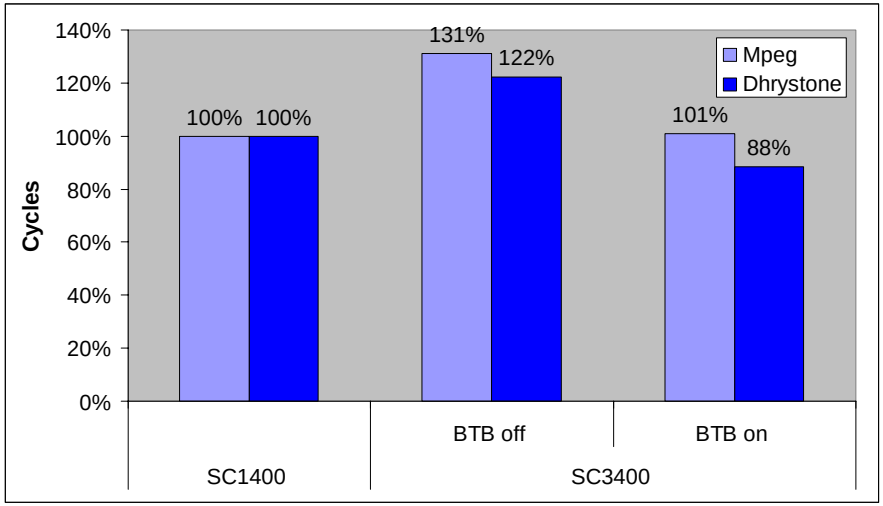
```
move (r0),d3  ; (r0)→ d3
mac  d3,d5,d6 ; d6 +(d3.H * d5.H)→ d6
move d6,(r7)  ; d6 → (r7)
```



Program Flow Features

- Minimization of latencies
  - Dynamic branch prediction
  - SP pre-calculation
  - Speculation
  - Zero overhead loops
  - Predication

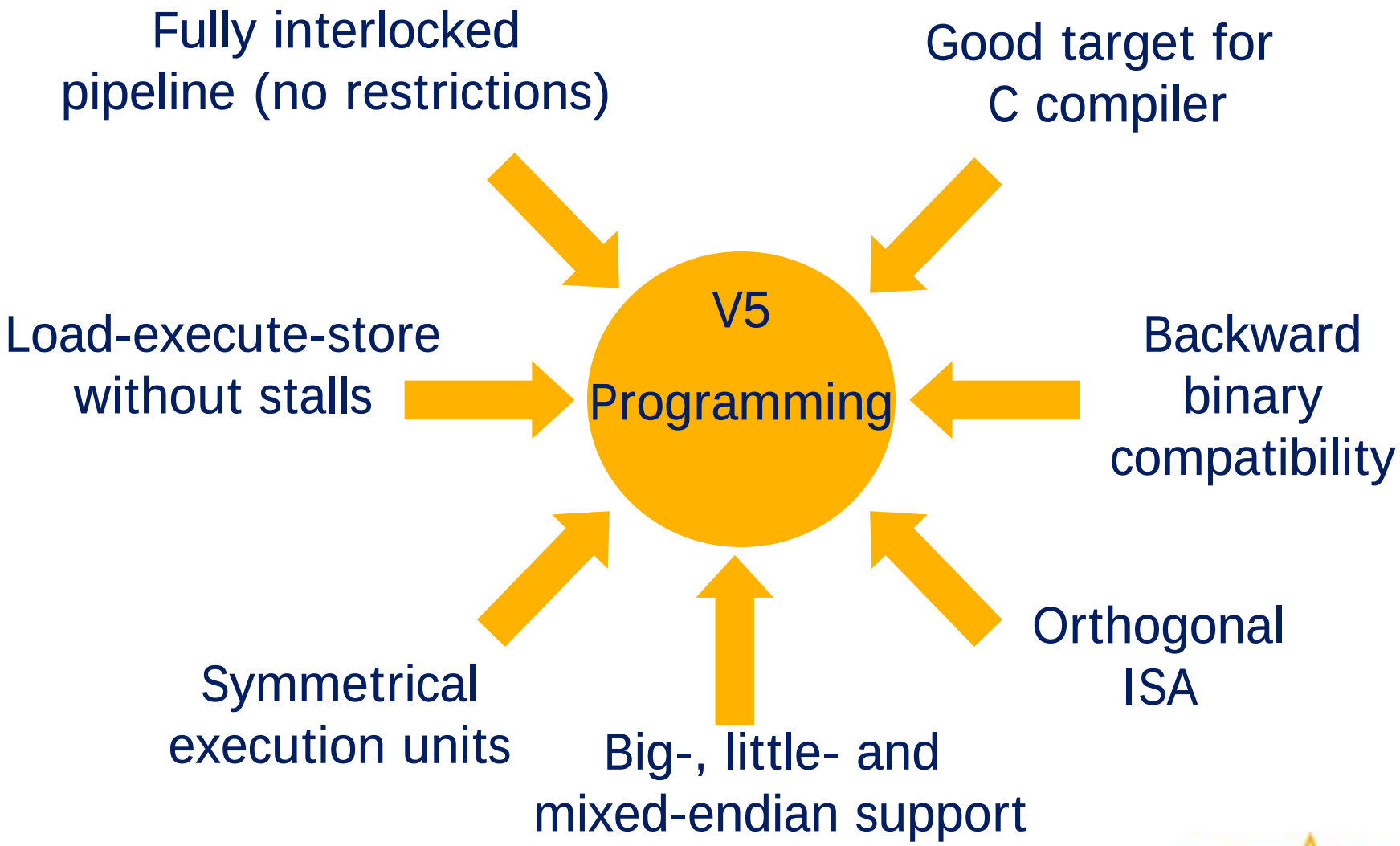
Legacy binary code performance



Instruction	Number of cycles SC1000	Number of cycles SC2000		Number of cycles SC3000		Comments
		BTB[(1)] hit	BTB miss	BTB hit	BTB miss	
Jump or Branch Conditional	1 or 4	1	4-5	1	6-11	SC1000 uses static branch prediction, chosen by user (BF, BT) SC2000 and SC3000 use dynamic branch prediction (automatically done by the processor)

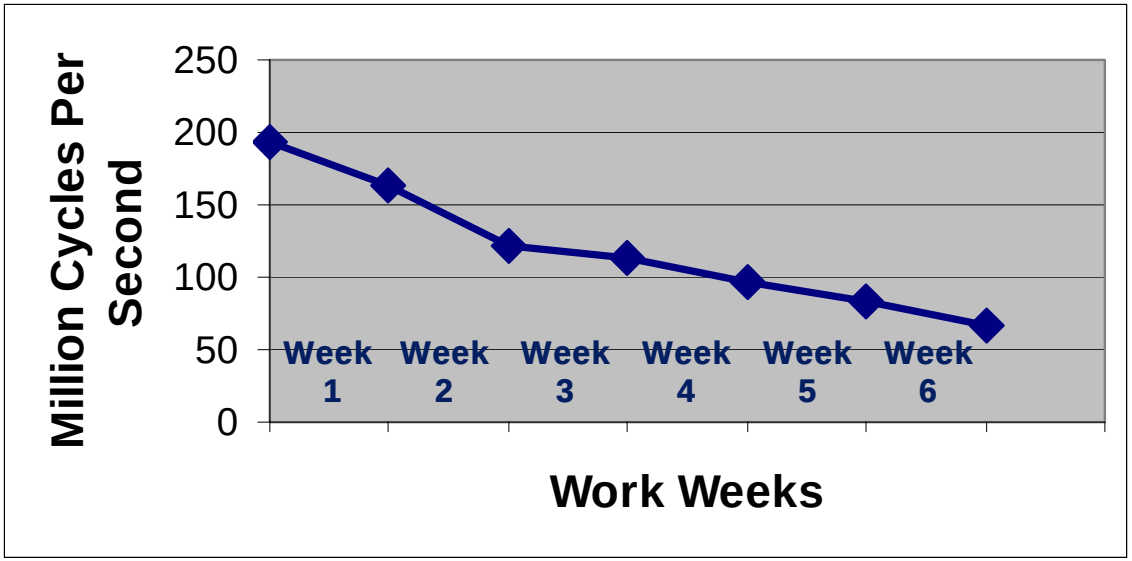
(1) Branch Target Buffer (BTB) is a cache used by the processor to perform the dynamic branch prediction

Focus - Ease of Programming



Programming Ease Example: H.264 Development on V5

- Application
  - H.264 decode, CIF 30 fps, 384 Kbps
- Starting point
  - C-compiled code using standard libraries
- Optimization flow
  - Integration of StarCore's video library
  - Code optimization



Source: Customer code optimized by StarCore

Features Enhancing MCU Capability

- Control application features
  - Control oriented instructions
  - Byte orientation
  - Reduced pipe break latency
  - Task protection (supervisor/user mode)
  - MMU Support
  - ALU operation in the AGU
  - Wide stack operations
- Code density approaching ARM Thumb®
  - New instructions for improving code density
  - Architecture and implementation features (e.g. fully interlocked pipe)

	ARM 9 Thumb [1]	SC3400 [2]	SC3400 vs. ARM 9 Thumb
GSM/GPRS Protocol Stack	**303kb**	**318kb**	**+5%**
ECC 1.2.1 (Open Source)	**492b**	**522b**	**+6%**

[1] ARM RCVT2.0.1 [2] SCLLC Compiler V1.2.5

Signal Processing and Application Specific support

- Application-specific instructions to accelerate video and digital communications applications
  - Video: motion estimation, deblocking, interpolation, etc
  - Communications: Viterbi decoders
- Additional instructions and features to accelerate general signal processing and bit manipulation functions like:
  - FDCT
  - FFT
  - 3D transport layer functions
  - Graphics

New Instructions

New Instructions by Functional Group	New Opcodes	Benefits
Compiler and RTOS support	16	■ Faster context switch ■ Lower cache miss penalties ■ Support for precise exceptions ■ Reduced code size ■ Faster control code
SIMD instructions and half-word support	19	■ Process twice the number of operands in parallel ■ Increase amount of registers by adding half register operands
Multimedia	6	■ Multimedia acceleration
Other application-specific instructions	6	■ Faster Viterbi algorithms ■ Faster interleaving/deinterleaving of bit streams
Total	47	

Viterbi Accelerator Not Needed

Application	Constraints length	Rate	Bit rate	Cycles Per Second (in Millions)	
				V2, V3, V4	V5
GSM voice channel	5	1/2	12.2Kbps	0.12 MCPS	0.05 MCPS
3G voice channel	9	1/3	12.2Kbps	2.4 MCPS	0.95 MCPS
EGPRS data channel	7	1/2	384Kbps	19.5 MCPS	7.8 MCPS

Viterbi accelerator is not needed

```
inner_loop:
   loopstart1
         [  sod2asxx d13,d13,d13          ; d13.H= x-y= A, d13.L= x+y= B
         ]
         [  acs2H -d14.L, d14.L,d4,d0     ; S0,S8,A        S0,S1
            acs2L  d14.H,-d14.H,d0,d4     ; S1,S9,B        S2,S3
            acs2H -d14.L, d14.L,d5,d1     ; S2,S10,A       S4,S5
            acs2L  d14.H,-d14.H,d1,d5     ; S3,S11,B       S6,S7
            move.b btr1.hh,(r2)-
         ]
         [  acs2H  d14.L,-d14.L,d6,d2     ; S4,S12,-A      S8,S9
            acs2L -d14.H, d14.H,d2,d6     ; S5,S13,-B      S10,S11
            acs2H  d14.L,-d14.L,d7,d3     ; S6,S14,-A      S12,S13
            acs2L -d14.H, d14.H,d3,d7     ; S7,S15,-B      S14,S15
            move.b btr1.hh,(r2)-
            move.l (r4)+,d15              ; read next x,y ,d15.H= x, d15.L= y
         ]
         [  sod2asxx d15,d15,d15          ; d15.H= x-y= A, d15.L= x+y= B
         ]
         …
```

Significant Power Improvement

- Power as a design target during architecture and implementation stages
- Power improvement methods
  - Relaxed memory timing
    - Enables use of high-density memories
  - Improved fetch-ahead algorithm
    - Fewer power-consuming mispredictions
  - Improved branch prediction
  - Shutdown of unused logic
    - Unused logic does not toggle
  - In-lane data access

Memory Density Example

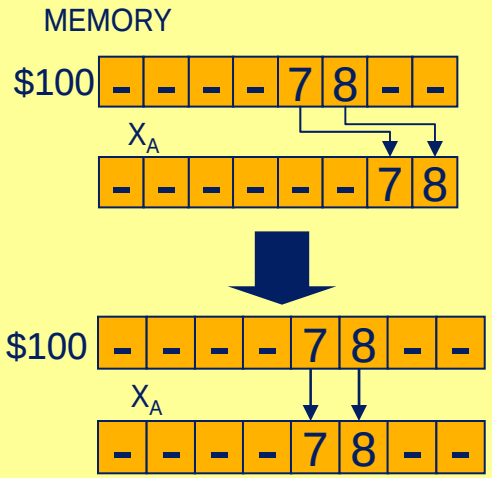
SRAM 2048 x 32 bit, Artisan, TSMC, 90nm G

high speed

high density

52% less power

SC3000 in-lane access



Summary

- StarCore V5 architecture offers:
  - New multimedia instructions to enable software implementation of numerous video standards
  - Enhanced MCU capability with features such as precise exception, wide stack operation and compelling code density
  - Many new compiler-friendly features
  - Backward binary compatibility
- StarCore SC3000 core offers improvement in:
  - Power consumption, frequency, code density and cycle count per application


StarCore™